

华中科技大学

2022

嵌入式系统

课程实验报告

题 目:	香橙派相关实验
专 业:	计算机科学与技术
班 级:	CS2006
学 号:	U202015471
姓 名:	杨释钧
电 话:	18790681696
邮 件:	y777x7@hust.edu.cn

华中科技大学课程设计报告

目 录

1	实验一 系统烧录	3
1.1	实验要求	3
1.2	实验过程	3
1.3	实验结果	3
2	实验二 图形界面基础.....	4
2.1	实验要求	4
2.2	实验过程	4
2.3	实验结果	4
3	实验三 图片文字显示.....	6
3.1	实验要求	6
3.2	实验过程	6
3.3	实验结果	6
4	实验四 多点触摸开发.....	8
4.1	实验要求	8
4.2	实验过程	8
4.3	实验结果	8
5	实验五 蓝牙通讯	10
5.1	实验要求	10
5.2	实验过程	10
5.3	实验结果	10
6	实验六 综合实验	12
6.1	实验要求	12
6.2	实验过程	12
6.3	实验结果	13

华中科技大学课程设计报告

7	实验总结与建议	15
7.1	实验总结	15
7.2	实验建议	15

1 实验一 系统烧录

1.1 实验要求

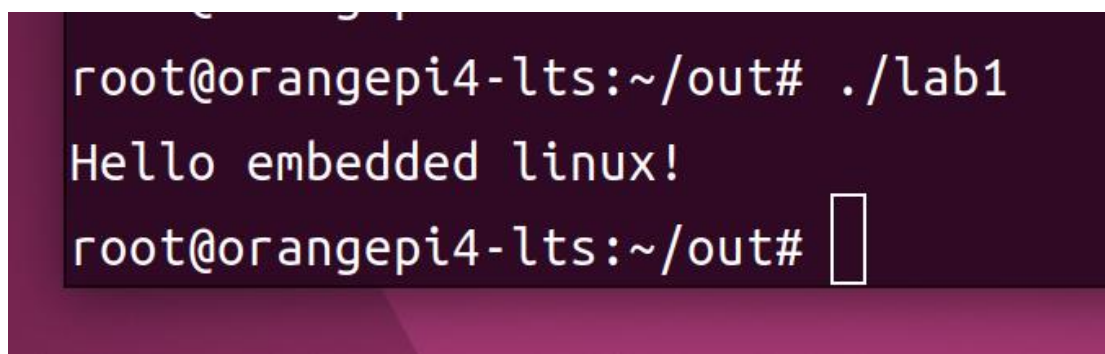
1. 连接开发板；
2. 烧录香橙派系统镜像；
3. 对香橙派进行网络配置；
4. 在香橙派上进行简单 Linux 应用开发。

1.2 实验过程

首先根据实验指导书连接开发板，然后用 balenaEtcher 对系统映像进行烧录，然后对香橙派进行网络配置，并且配置本地编译环境，最后编译运行 lab1 即可。

1.3 实验结果

运行结果如图 1.1 所示。



```
root@orangepi4-lts:~/out# ./lab1
Hello embedded linux!
root@orangepi4-lts:~/out#
```

图 1.1 实验一运行结果

2 实验二 图形界面基础

2.1 实验要求

1. 了解 Linux 下的 LCD 显示驱动接口：framebuffer 的使用原理；
2. 理解 Linux 下基本图形的显示，如点、线、矩形区域等；
3. 理解双缓冲机制；
4. 实现画矩形函数 `fb_draw_rect` 和画线函数 `fb_draw_line`，最终绘制出需要的图形；

2.2 实验过程

2.2.1 画矩形函数 `fb_draw_rect` 的实现

该函数的实现比较简单，可以利用预先实现好的画点函数 `fb_draw_pixel`，用一个两重循环调用画点函数将需要构成矩形区域的位置进行填充即可。

2.2.2 画线函数 `fb_draw_line` 的实现

该函数的实现相对复杂一点，由于本身该函数的前四个参数给出了所需要画线的起点和终点，如果仅仅使用直线点斜式方程，利用两点坐标计算斜率，然后进行循环连线的话，会导致画线精度不够，最后在斜率偏大的时候直线会变成一个一个的点。因此，为了保证最终得到的直线尽可能光滑，我们根据待画线的斜率进行区分，当待画线斜率绝对值小于 1 的时候，根据横坐标的值去计算纵坐标，也就是当横坐标每次增加 1 的时候去计算此时纵坐标的值，然后画点；否则只需要根据纵坐标的变化进行画点即可。

2.2.3 运行与调试

这里主要出现了两个问题：

1. 画线最终的结果不够光滑。这是采用的画线的策略导致的，在 2.2.2 节中已经提到过；
2. 显示的图形无法充满整个显示屏。经过测试发现是程序中原有的关于显示屏尺寸的宏 `SCREEN_WIDTH` 和 `SCREEN_HEIGHT` 出了问题，原因是我最开始将开发板连接到了实验室的显示器上，后续才将开发板连接到了配套的显示器上，只需要重启一下开发板即可。

2.3 实验结果

可以看到，最终实验二的运行结果如图 2.1 所示。性能测试相关部分将在实

验三中展示。

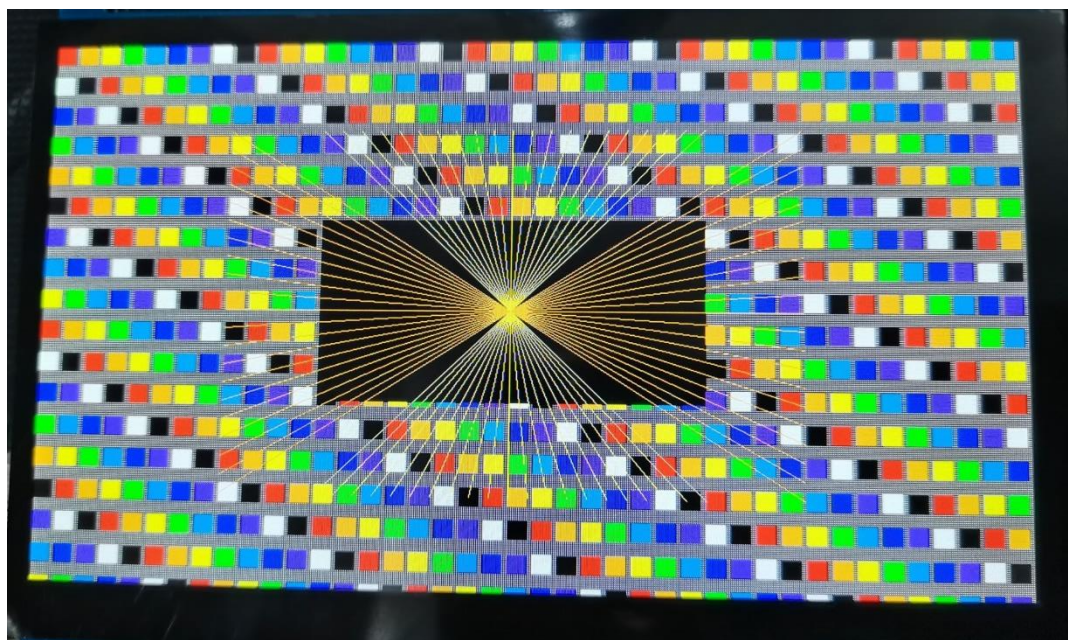


图 2.1 实验二运行结果

3 实验三 图片文字显示

3.1 实验要求

1. 显示 jpg 不透明图片；
2. 显示 png 半透明图片；
3. 显示矢量字体，需要完成对字模的提取和对字模的显示。

3.2 实验过程

3.2.1 画图函数 fb_draw_image 的实现

本实验主要需要实现画图函数 fb_draw_image 来实现对 jpg 不透明图片、png 半透明图片和矢量字体的显示，然后对画图函数在性能上进行一定的优化即可。这一小节将介绍 fb_draw_image 实现的主要思路。

根据代码框架，当图片的颜色类型 color_type 成员为 FB_COLOR_RGB_8880 时，说明此时需要显示的是 jpg 不透明图片，也就是不需要对透明度进行处理，只需要将图片内容复制到缓冲区即可；当图片的颜色类型 color_type 成员为 FB_COLOR_RGB_8888 时，说明此时需要显示的是 png 半透明图片，需要利用实验任务书中给出的关于透明度 alpha 和 RGB 的存储方式进行读出，并且利用任务书中给出的公式进行重新计算即可；当图片的颜色类型 color_type 成员为 FB_COLOR_ALPHA_8 时，说明此时需要显示的是矢量字体，其处理方法和 png 半透明图片类似，不同的是需要根据函数参数进行设置颜色。

3.2.2 性能调优

这里主要采取了一些比较简单地性能优化方法，比如更换透明度的运算顺序来减少乘除法指令的次数、利用 memcpy 函数来对一整块内容进行复制从而获取一个比较好的性能。

3.3 实验结果

可以看到实验三运行结果如图 3.1 所示。



图 3.1 实验三运行结果

性能测试结果如图 3.2 所示。

嵌入式系统实验一测试	
操作	时间 (ms)
点pixel	13
矩形rect	20
线line	35
图像image	121
文本text	10
合计total	199

华中科技大学

图 3.2 性能测试结果

4 实验四 多点触摸开发

4.1 实验要求

1. 了解 Linux 下的触摸屏驱动接口原理，了解 Input event 的使用，了解多点触摸协议（Multi-touch Protocol）；
2. 获取多点触摸的坐标；
3. 在 LCD 上显示多点触摸轨迹，要求可以使用多根手指在屏幕上进行绘图，每根手指的轨迹应该不同，轨迹应该是连贯的，不能断断续续，轨迹应该比较粗，线宽不能是一个点，绘制一个清除屏幕的按钮，点击后清除屏幕内容。

4.2 实验过程

根据任务书中的指示，我们主要需要填充函数 `touch_event_cb`，也即需要对 `TOUCH_PRESS`、`TOUCH_MOVE`、`TOUCH_RELEASE`、`TOUCH_ERROR` 四个事件做出相应的响应即可。`TOUCH_PRESS` 事件表示当前手指首次触碰到屏幕，这个时候需要存储相关的信息，并且判断是否需要进行清除操作；`TOUCH_MOVE` 事件表示当前手指发生了移动，需要进行画线；`TOUCH_RELEASE` 事件表示手指离开了屏幕，不需要进行特殊处理；`TOUCH_ERROR` 事件表示触碰异常，需要终止程序。

首先，为了满足实验要求中的轨迹较粗的特点，我们这里自己实现了一个画线函数，该函数画出的线较粗，实现思路是利用画圆的函数，将从线的起点和终点之间的每一个像素点为圆心画圆，这样得到的线条就会比较粗。

为了实现画线，我们需要保存手指曾经的位置，这部分用一个数组去存储即可。为了实现每个线的颜色不同，用一个数组来预存储待用到的颜色，然后在需要画线的时候根据历史信息选择绘画的颜色即可。

最后，为了实现清除屏幕的按钮这一功能，我们需要设定一个清屏按钮的区域，这里我选择的是左上角的正方形区域，如果触碰到这个区域的话，只需要将整个屏幕全部涂白，然后将清屏按钮涂黑即可，实际上相当于重新进行了一次初始化操作。

4.3 实验结果

这里展示一下屏幕绘图功能，如图 4.1 和图 4.2 所示。



图 4.1 实验四部分运行结果

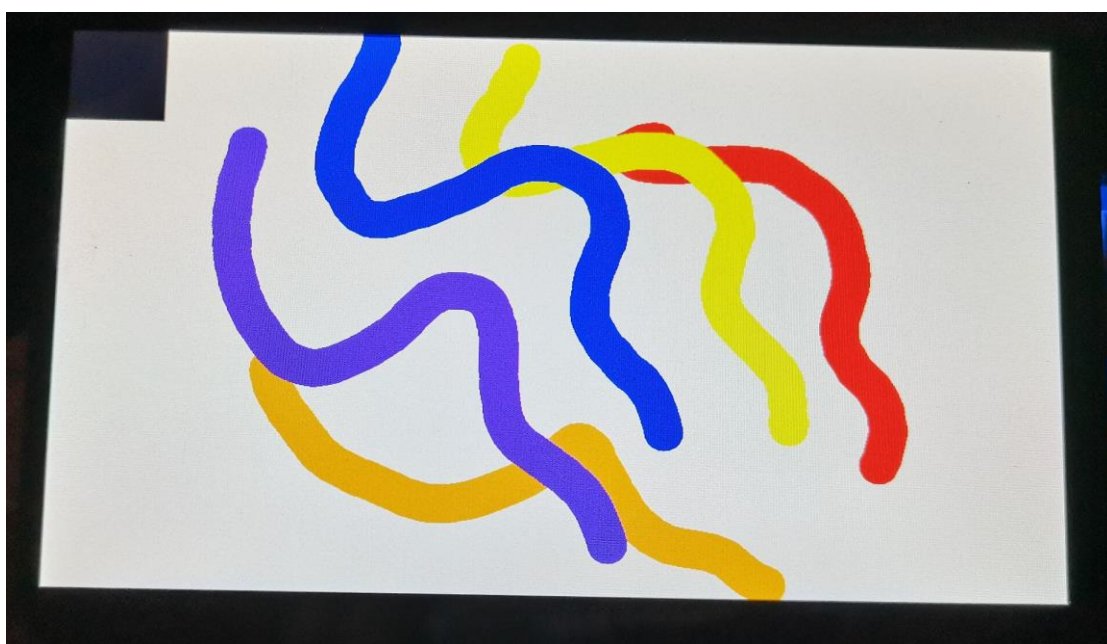


图 4.2 实验四部分运行结果

5 实验五 蓝牙通讯

5.1 实验要求

1. 正确配置并启动蓝牙服务；
2. 熟练使用蓝牙的常用命令工具；
3. 通过 RFCOMM 串口（蓝牙串行通讯）实现无线通信，测试通过 lab5 实验代码；

5.2 实验过程

5.2.1 配置并启动蓝牙服务

首先编辑开发板中的配置文件，随后利用 `bluetoothctl` 进入蓝牙 shell，使用命令 `power on`、`discoverable on`、`pairable on` 开启相关配置，然后将手机和开发板的蓝牙进行配对。

5.2.2 测试 lab5

在手机上安装 SPP 蓝牙串口程序，开发板运行 `rfcomm -r watch 0 1`，随后将开发板和手机蓝牙建立连接，开发板上运行 lab5 测试程序即可开始测试。

5.3 实验结果

手机上 SPP 蓝牙串口程序界面如图 5.1 所示。



```
22:35:56.814> hello
22:35:56.938> hello
22:35:57.084> hello
22:35:57.311> hello
22:35:57.442> hello
22:35:57.664> hello
22:36:01.823> 杨释钧
22:36:13.765> YANG SHIJUN
22:36:21.261> 计算机科学与技术学院
```

图 5.1 SPP 蓝牙串口程序运行界面

开发板屏幕上显示的信息如图 5.2 所示。

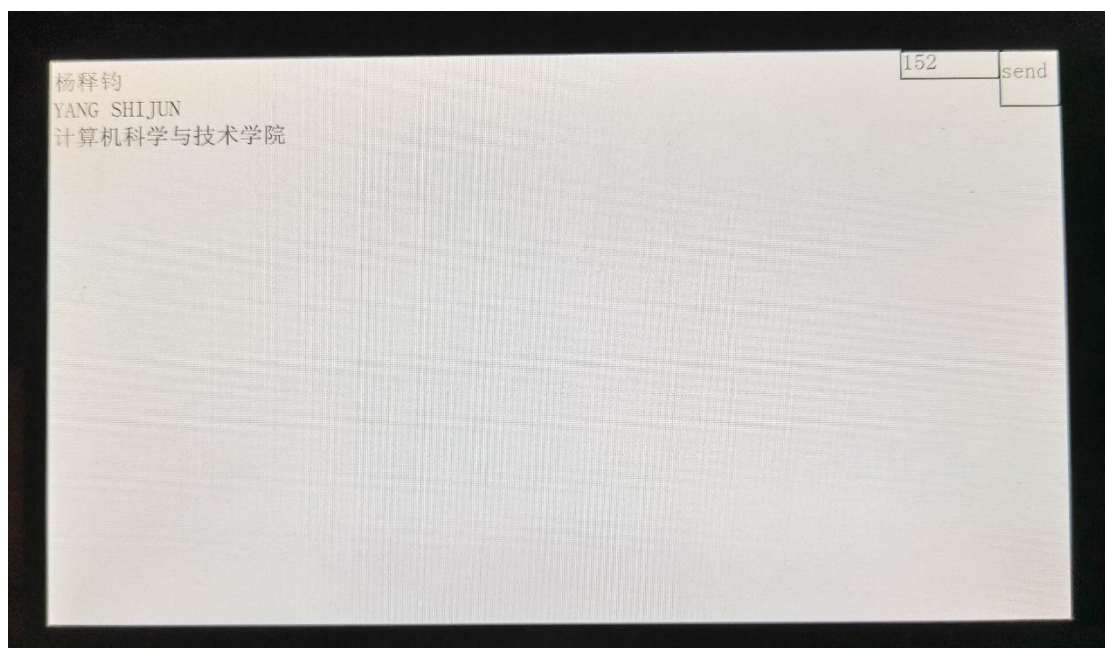


图 5.2 开发板显示器上的打印信息

开发板终端输出如图 5.3 所示。

```
bluetooth tty send hello
bluetooth tty send hello
bluetooth tty send hello
bluetooth tty receive "杨释钧"
bluetooth tty receive "YANG SHIJUN"
bluetooth tty receive "计算机科学与技术学院"
close bluetooth tty fd
root@orangepi4-lts:~/out#
```

图 5.3 开发板终端输出

6 实验六 综合实验

6.1 实验要求

综合运用所学内容，设计并实现一个在多个开发板之间进行蓝牙互联协作的程序或一个功能比较复杂的单机程序。例如：

1. 共享白板：两个开发板之间共享屏幕手绘内容；
2. 共享文件：显示远程开发板上的目录文件列表, 选中指定文件，显示指定文件内容(文本和 png/jpg 图片)；
3. 语音消息：两个开发板之间语音转文字并发送到对方屏幕；
4. 联网游戏：比如五子棋等；
5. 单机程序：
 - 1) 图片浏览器：图片放大、缩小、图片尺寸超过显示区域时手指拖动显示；
 - 2) 视频播放器；
 - 3) 语音识别程序：语音控制屏幕显示；
 - 4) 游戏或其他；

6.2 实验过程

这里选择的是共享白板，当然为了增加一定的工作量，在最简单的共享白板功能的基础上增加了一些功能，比如橡皮擦、调整画笔颜色、调整画笔粗细等等。

实现共享白板这一功能的核心思路参考了 lab4、lab5，白板的绘画功能类似于 lab4，而无线通信参考了 lab5，也就是一方面完善 touch_event_cb 使其支持上述提到的一些功能，如调整画笔颜色等等，另一方面完善 bluetooth_tty_event_cb 函数，来实现无线通讯。

为了通过共享白板实现无线通讯，需要利用 myWrite_nonblock 函数和 myRead_nonblock 函数进行数据传递，也就是说共享白板的两端都需要有本地屏幕的事件以及对蓝牙进行响应的事件，在本地接收到响应之后同时也需要将这次事件对应的信息传递给另外一段，从而实现共享的功能。

此外，这里也参考了 lab3 中的一些知识，增加了一些图片来使得本实验的功能更加美观。

6.3 实验结果

下面对上文中提到的一些功能进行测试，首先是调节画笔粗细的功能，如图 6.1 所示。

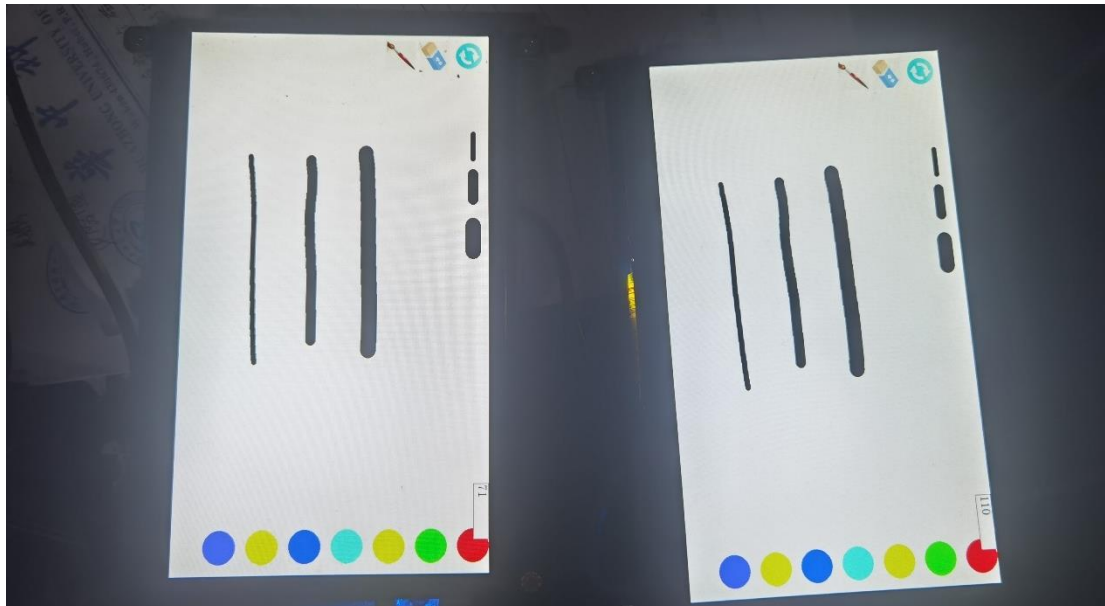


图 6.1 调节画笔粗细功能测试

然后是调节画笔颜色的功能，如图 6.2 所示。

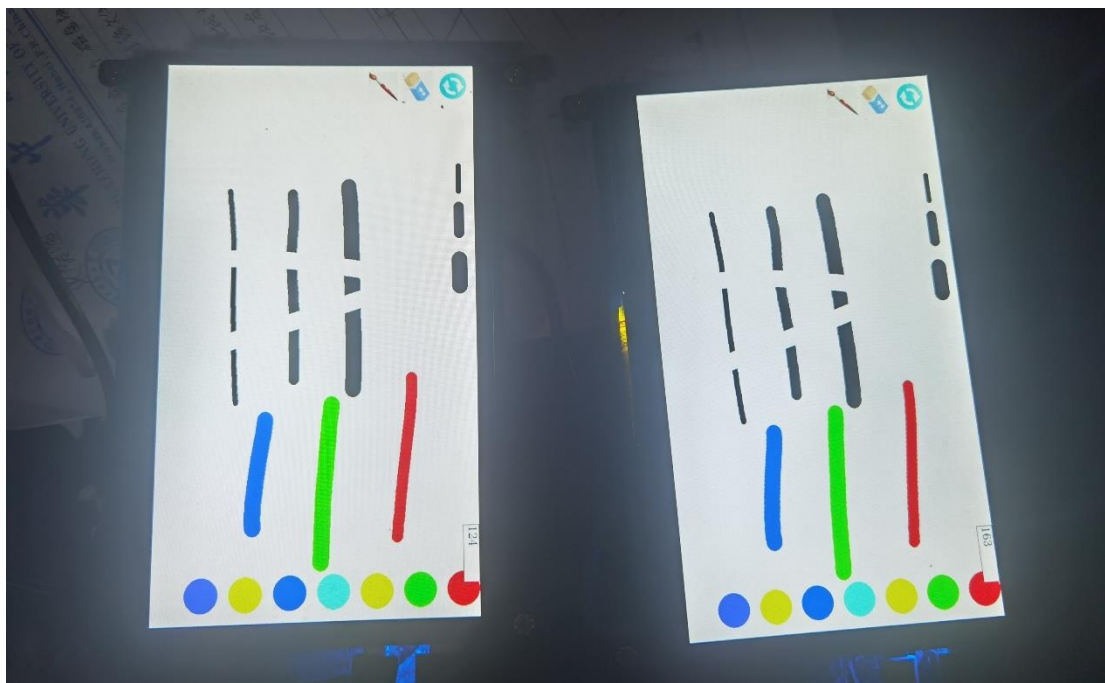


图 6.2 调节画笔颜色功能测试

最后是橡皮擦的功能，如图 6.3 所示。

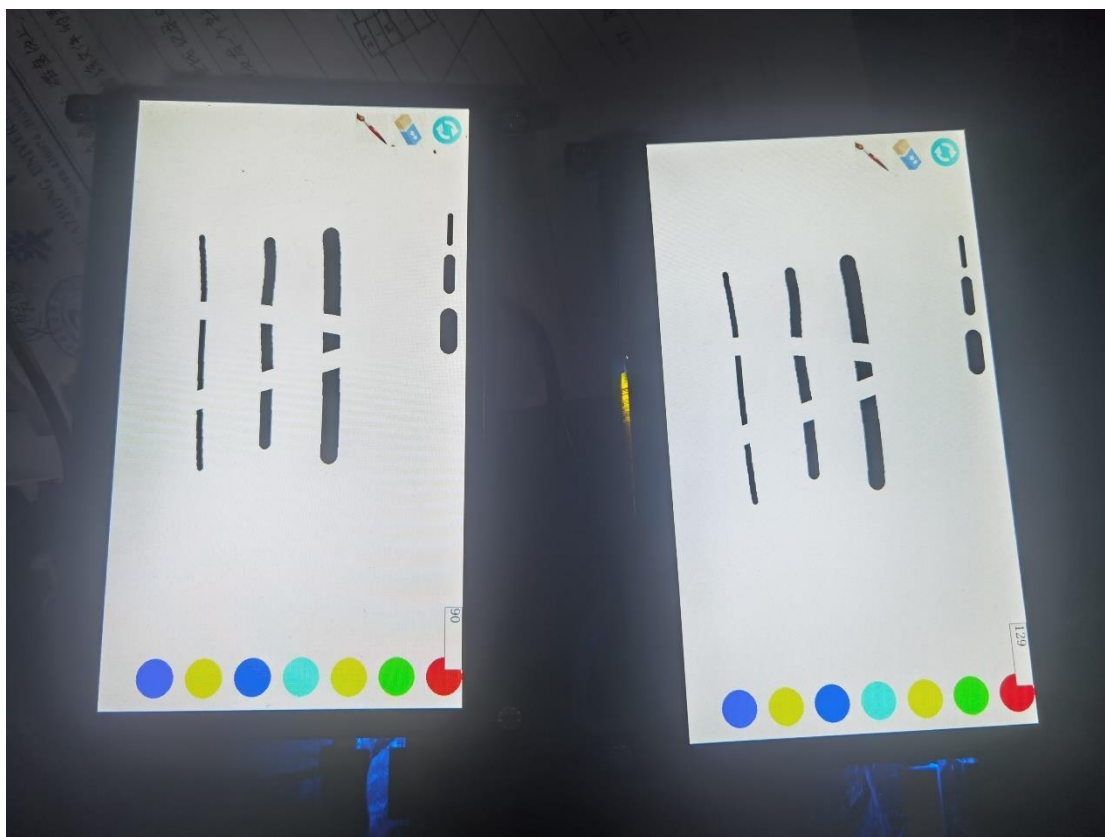


图 6.3 橡皮擦功能测试

7 实验总结与建议

7.1 实验总结

通过本次实验，我共完成了香橙派的 6 个 lab，在实践中学习了嵌入式系统开发的基本知识与方法。

在 lab1 中，我根据实验指导书成功将系统烧录到开发板中，并且通过网络配置用 ssh 远程连接到了开发板，并且成功运行了一个简单的 Linux 应用程序。

在 lab2 中，我学习了图像显示的双缓冲机制的原理，并且成功实现了一些基本图形的绘制，如矩形、线段等，同时也在老师的指导下学习了在嵌入式开发中寻找潜在 bug 的一些方法和手段。

在 lab3 中，我学习了图像存储和显示的一些原理，对 Linux 下图形界面的显示有了更深层次的了解，同时也在老师的指导下学习了一些性能调优的方法。

在 lab4 中，我学习了 Linux 下多点触摸协议的相关知识，并且在其基础上成功实现了绘制手指轨迹并且清除轨迹这一功能。

在 lab5 中，我学习了 Linux 设备利用蓝牙进行无线通讯的原理，并且在任务书的指导下成功将开发板和手机的蓝牙串口设备进行互联，成功完成实验。

在 lab6 中，我综合了前面的几个实验，基于蓝牙设计了共享白板的实现，对 Linux 基于蓝牙的无线通讯以及基于多点触摸协议的程序开发有了更为深刻的认知。

总的来说，通过本次课程的学习，我对嵌入式系统开发中的一些知识有了深刻了解，比如基于蓝牙的无线通讯、Linux 下图像显示的一些原理等等，收获颇丰。

7.2 实验建议

本次课程设计的实验我个人感觉难度适中，很适合作为嵌入式系统学习的入门实验，下面给出几个我个人的小小的建议：

1. 在 lab1 中配置开发环境的部分，其实比较建议增加用 vscode+ssh 进行远程开发的部分，现在这样的工具已经做的比较成熟了，不过这样可能少了在本地配置交叉编译环境这样一个步骤；

2. 可以把实验任务书从 PPT 的形式改成 PDF 的形式，可能会显得更加正式一点，并且也能更加详细的介绍更多的知识。